AUTOMATIC TROUBLE ANALYSIS (ATA) SYSTEM
COMPUTER INPUT COMMANDS/OUTPUT MESSAGES

CONTENTS PAGE

1. GENERAL

1.01 This section provides the input commands/output messages used by personnel at the Automatic Trouble Analysis (ATA) System console.

1.02 This section is reissued to include additions and changes to input commands and output messages for Generic 3. Revision arrows are used to emphasize the more significant changes.

1.03 The following conventions are used in the description of computer input commands/output messages:

(a) Brackets [ ] are used to enclose optional elements of a command.

(b) A space indicates that one or more spaces must appear in that position.

(c) When the user must respond by operating a particular key or combination of keys, the key name is bracketed by < >.

(d) All input commands are ended by operating either the carriage return (CR) or line feed (LF) key.

NOTICE
Not for use or disclosure outside the
Bell System except under written agreement

2. CONSOLE INPUT

2.01 The system console is the direct access to UNIX* operating system, which operates with the following conventions:

(a) The $ or # symbol informs the user that UNIX operating system is ready for an input command.

(b) Typing errors can be corrected by using the # key or the @ key. Each use of the # key causes the previous typed character to be deleted; eg, if the # key is operated twice, the two preceding characters will be deleted. The @ key deletes the entire line of typing up to the @ symbol.

(c) A line of input is ended by operating the CR or LF key.

(d) To terminate a program, type either RUBOUT (DEL on some terminals) or CTRL-D.

2.02 When the system is started or when the user logs off, UNIX operating system requests that the user type in an identification by printing **LOGIN:**. The user must respond by typing an identification, followed by either CR or LF. If the identification given is valid, UNIX operating system responds with a prompt character ($ or #) possibly preceded by a logic message. The user can now type in the name of a program to be run.

2.03 If the user gives an invalid identification, UNIX operating system prints an error message and requests a new identification:

LOGIN INCORRECT

LOGIN:

The user must respond with a valid identification.

2.04 Occasionally UNIX operating system refuses to accept a valid identification. If this occurs, log off (paragraph 2.05) and try again. UNIX operating system may also decide that your terminal cannot display both uppercase and lowercase characters and use only uppercase when printing to you. If you do have an uppercase/lowercase terminal and this happens, log off and login again.

2.05 To log off a terminal, type **CTRL-D** (twice if no response occurs the first time). UNIX operating system responds with ;**LOGIN:**.

2.06 To change the current login identification at the console, the user can either log off and log back in or type **LOGIN** and identification after a prompt.

3. INPUT COMMANDS/OUTPUT MESSAGES

A. Input Commands

3.01 Each input command entry consists of four parts of information which are as follows:

(a) Input command format shows the format for the particular command.

(b) Description gives a detailed explanation of the command function and identification of any variable fields.

*Trademark of Bell Laboratories.

(c) Examples show the system response to the input command.

(d) Where applicable, references list any complementary commands.

3.02 A list of the input commands is given in Table A.

TABLE A

INPUT COMMANDS

COMMAND	FORMAT	PAGE
Generic 3: Add Alarm Monitor	addalmmon	7
Generic 3: Add Backup Site	addbackup	7
Add Monitor Site	addmon	7
Add Office	addoffice	8
Add Options	addopt siteid Generic 3: addopt siteid key card	10
Askcodes	askcodes siteid 5 key Generic 3: askcodes siteid key card	10
Generic 3: ATA	ata	11
Backup	backup [epoch, savei, geti]	12
Generic 3: Backup Monitor Site	backupmon	13
Generic 3: Broadcast	broadcast "text"	14
Generic 3: Browse	browse browse filename b b filename	14
Change Directory	chdir directory cd directory	15
Check	check [-sn][file system]	15
Generic 3: Change history	chghist sitename	19

TABLE A (Contd)

INPUT COMMANDS

COMMAND	FORMAT	PAGE
Compile	compile siteid key Generic 3: compile siteid key card	19
Concatenate	cat[-u] file...	15
Generic 3: Convert	convert key data conv key data	20
Copy	cp [-t] <fname> <ufname>	21
Crossbar Type	xbtype key	46
Date	date [mmddhhnn][yy]	21
Direct Actions of Init	telinit [arg]	40
Generic 3: Dump Error Message	dmpem tc file	22
Generic 3: Dump ETS	dmpets tc file	22
Generic 3: Dump MDT	dmpmdt tc file	23
Generic 3: Dump OTF	dmpotf tc file	23
Generic 3: Dump TPIU	dmptpiu tc file	23
Generation Options	genopt sitename key1 key2 Generic 3: genopt sitename key1 card	23
Install UNIX Shell (Not used in Generic 3)	unixshell	42
Line Control	lctrl siteid key option	24
List	ls [-1tasdruifI][filesys] name ...	28
Load Card	ldcd sitename Generic 3: ldcd sitename key card	26

TABLE A (Contd)

INPUT COMMANDS

COMMAND	FORMAT	PAGE
Loadrun	loadrun key	27
Generic 3: Lograw	lograw [site/line][name/number] key	27
Loop	loop lnxx lnyy [speed] Generic 3: loop lnxx lnyy [baud rate]	28
Make File Section	/etc/mkfs special proto [-b[numbers]]	29
Make Fixed Head Disk (not used in Generic 3)	makefh	29
Make Node	/etc/mknod/dev/lnxx c[4 6] xx	29
Mount	/etc/mount special file [-r]	30
Move	mv <fname1> <fname2>	30
Office Type	oftype name key card Generic 3: oftype name key card (generic/issue)	31
Generic 3: Output Pattern	oppat pname	32
Password	passwd name password	33
Generic 3: Pattern Generation	patgen pname	33
Remove (not used in Generic 3)	remove siteid. key	33
Generic 3: Remove Alarm Monitor	rmalmmon	35
Remove Entries From Files	rm [-f][-r][-d] name...	34
Generic 3: Remove Backup	rmbackup	35

TABLE A (Contd)

INPUT COMMANDS

COMMAND	FORMAT	PAGE
Remove Directory	rmdir [-r] directory	35
Remove Site	rmsite sitename	35
Restore (not used in Generic 3)	restore siteid. key	34
Save Altered Data	rtdsave Generic 3: rtdsave sitename	36
Shell	sh [-tcvi] [name (arg1...arg9...)] Generic 3: sh [-ecnpstvx] [arg...]	39
Stand Alone Copy	sacopy	37
Start ATA	startata [init] startata [noinit] startata [erdsave]	39
Stop ATA	stopata	39
Sync	sync	40
System Answer	sysans officename key	40
Terminal Control	termcntl	40
Transfer (not used in Generic 3)	transfer lnxx lnyy	42
Unmount	/etc/umount special	42
Update File System	/etc/updfs [-nxctviogf] [name] [file] [dir] Generic 3: /etc/updfs [-nmxctviogfue] [tapefile] [name] [file] [dir]	43
Verify Line	vrfyline [line]	44
Verify Site	vrfysite [sitename]	44
Yesno	yesno sitename key	46
Zaptables	zaptables	47

♦Generic 3 with PDP*-11/70: ADDALMMON

A. Description: The addalmmon command allows a line to be defined for the watchdog monitor function for the ATA system. The line is set for 1200 baud.

B. Example: Add line number serving watchdog monitor.

```
ADDALMMON
```

System response:

```
LINE NUMBER SERVING ALARM MONITOR♦
```

Generic 3 with PDP-11/70: ADDBACKUP

A. Description: The addbackup command allows lines to be defined as dial backup for ETS offices. The line must be served with a 212A data set and will be set up for 1200 baud.

B. Example: Add line 37 to ETS dial backup.

```
ADDBACKUP
```

System response:

```
BACKUP LINE NUMBER? <37>
LINE 37 ASSIGNED TO ETS DIAL BACKUP
```

ADDMON

A. Description: The addmon command defines a monitoring site to the ATA System. The program interactively obtains the name, line number, and system power for the office being defined. For Generic 3, the program also obtains mode of operation [Telecommunications Alarm Surveillance and Control (TASC), UNIX operating system, or Electronic Translation System (ETS)] and mode default for the office being defined. After all of the interactive information has been obtained and checked for reasonableness, the system distribution, reach, and command acceptance matrices are updated along with the site data table, office names table, and line translation table.

B. Examples:

1. Add a monitoring site to line 12.

```
ADDMON
```

System response:

```
ENTER SITE NAME - <CNTRVILLE>
LINE NUMBER SERVING TERMINAL - <12>
IS THIS A MULTIPLEXED (MDT) LINE? (Y-N) - <Y>
IS THIS THE SECOND TERMINAL ON A NO. 1 CROSSBAR MDT? (Y-N) - <N>
```

*Registered trademark of Digital Equipment Corporation.

```
ENTER COMMAND POWER OF SITE - CHOICES ARE SCC OR SUBSCC - <SCC>
ENTER BAUD RATE OF SITE TERMINAL - CHOICES ARE 300 AND 1200 - <1200>
DEFINITION COMPLETE FOR CNTRVILLE, SITE 63
```

Generic 3:

2. Add a monitoring site to line 12 with TASC, UNIX operating system, and ETS mode.

ADDMON

System response:

```
ENTER SITE NAME - <KEYPORT>
ENTER COMMAND POWER OF SITE - CHOICES ARE SCC OR SUBSCC - <SCC>
LINE NUMBER SERVING TERMINAL - <12>
IS THIS THE SECOND TERMINAL ON A NO. 1 CROSSBAR MDT? (Y-N) - <N>
ENTER BAUD RATE OF SITE TERMINAL - CHOICES ARE 300, 1200, ▶1800, or 2400◀ - <1200>
IS TASC MODE REQUIRED? (Y-N) - <Y>
IS UNIX MODE REQUIRED? (Y-N) - <Y>
IS ETS MODE REQUIRED?  (Y-N) - <Y> (PDP-11/70 only)
ENTER TERMINAL TYPE - CHOICES ARE NONE, VT100, OR DS40-1 - <VT100>
ENTER MODE DEFAULT OPTION - CHOICES ARE ATA OR UNIX - <UNIX>
DEFINITION COMPLETE FOR KEYPORT, SITE 63
```

3. Add a monitoring site to line 23.

ADDMON

System response:

```
ENTER SITE NAME - <CNTRVILLE>
ENTER COMMAND POWER OF SITE - CHOICES ARE SCC OR SUBSCC - <SCC>
LINE NUMBER SERVING TERMINAL - <23>
IS THIS THE SECOND TERMINAL ON A NO. 1 CROSSBAR MDT?  (Y-N) - <N>
ENTER BAUD RATE OF SITE TERMINAL - CHOICES ARE 300, 1200, ▶1800, OR 2400◀ - <1200>
IS TASC MODE REQUIRED? (Y-N) - <N>
IS UNIX MODE REQUIRED? (Y-N) - <N>
IS ETS MODE REQUIRED? (Y-N) - <N> (PDP-11/70 only)
ENTER TERMINAL TYPE - CHOICES ARE NONE, VT100, OR DS40-1 - <VT100>
DEFINITION COMPLETE FOR CNTRVILLE, SITE 62
```

ADDOFFICE

A. Description: The addoffice command defines an office to the ATA System. The program interactively obtains the name, crossbar type, and ATA device type of the office being added. For No. 5 ETS offices, the addoffice program asks if ETS history logging is required. If ETS history logging is required, the program responds with the number of dedicated blocks, the number of used blocks, the number of available blocks, and a request for the number of blocks to be assigned for this office. The number of blocks assigned to an office is a function of how many days of data it is desirable to keep on temporary storage (disk) and the activity of the office. The addoffice program then builds the line translation table entries defining lines and message IDs for command and ATA information. After all interactive information has been obtained and checked for reasonableness, the system distribution, reach, and accept matrices are updated along with the site data table, office names table, and line translation table.

B. Examples:

1. Add XBOFF5 office to line 02.

```
ADDOFFICE
```

System response:

```
ENTER OFFICE NAME - <XBOFF5>
ENTER CROSSBAR TYPE - CHOICES ARE 1, 5, OR T - <5>
ENTER ATA DEVICE TYPE - CHOICES ARE MDT OR TPIU - <TPIU>
LINE NUMBER SERVING TTY - <02>
LINE NUMBER SERVING ATA DEVICE - <01>
CREATING OFFICE XBOFF5
DEFINITION COMPLETE FOR XBOFF5, SITE X
```

2. Add No. 1 crossbar office XBOFF1 to line 06.

```
ADDOFFICE
```

System response:

```
ENTER OFFICE NAME - <XBOFF1>
ENTER CROSSBAR TYPE - CHOICES ARE 1, 5, OR T - <1>
LINE NUMBER SERVING TTY - <06>
IS THIS THE PRIMARY MDT OFFICE SITE? (Y-N) - <Y>
DEFINITION OF MARKER GROUPS AND TROUBLE INDICATORS MAKING UP THE OFFICE:

IS THERE AN ORIGINATING MARKER GROUP? (Y-N) - <Y>
LINE NUMBER SERVING OTI - <06>
WHICH MARKER GROUP ON OTI? (0-1) - <0>
IS THERE A CTI FOR THIS MARKER GROUP? (Y-N) - <Y>
LINE NUMBER SERVING CTI - <06>
WHICH MARKER GROUP ON CTI? (0-1) - <0>
HOW MANY TERMINATING MARKER GROUPS? (0-3) - <1>

TERMINATING MARKER GROUP 0:
LINE NUMBER SERVING TTI FOR THIS MARKER GROUP - <06>
WHICH TTI ON THE LINE? (0-2) - <0>
WHICH MARKER GROUP ON THE TTI? (0-2) - <0>
HOW MANY ANI IDENTIFIER GROUPS? (0-3) - <1>

ANI IDENTIFIER GROUP 0:
LINE NUMBER SERVING ANI - <06>
WHICH IDENTIFIER GROUP ON ANI? (0-2) - <0>
IS THERE AN SSTI? (Y-N) - <N>
HOW MANY COMPRESSION INDICATORS? (0-2) - <1>
LINE NUMBER SERVING CCTI0 - <06>
IS THERE AN ASTI? (Y-N) - <N>
CREATING OFFICE XBOFF1
DEFINITION FOR XBOFF1, SITE X
```

Generic 3:

3. Add the XBOFF5 office to line 02.

```
ADDOFFICE

System response:
ENTER OFFICE NAME - <XBOFF5>
ENTER CROSSBAR TYPE - CHOICES ARE 1, 5, T, E, OR OTF5 - <5>
ENTER ATA DEVICE TYPE - CHOICES ARE MDT OR TPIU - <MDT>
LINE NUMBER SERVING XBOFF5 - <02>
CREATING OFFICE XBOFF5
DEFINITION COMPLETE FOR XBOFF5, SITE X
```

4. Add the XBOFF5E ETS office to line 04 (No. 5 ETS) with history logging.

```
ADDOFFICE

System response:
ENTER OFFICE NAME - <XBOFF5E>
ENTER CROSSBAR TYPE - CHOICES ARE 1, 5, T, E, OR OTF5 - <E>
LINE NUMBER SERVING XBOFF5E - <04>
IS ETS HISTORY LOGGING REQUIRED? (Y-N) - <Y>

DEDICATED BLOCKS: 49742, USED BLOCKS: 600
AVAILABLE BLOCKS:  49142

ENTER NUMBER OF BLOCKS: <500>
CREATING OFFICE XBOFF5E
DEFINITION COMPLETE FOR XBOFF5E, SITE X
```

▶Generic 3:

C. References: Chghist and rmsite commands.◀

ADDOPT siteid
Generic 3:
ADDOPT siteid KEY CARD
KEY = 1, 5, e ▶(PDP-11/70 only), or f◀
CARD = 3638, 4393, 5488, tind, or ▶xotf◀

A. Description: The addopt command reads the option file created by the genopt routine, translates field name to index numbers with the help of the card description source file, and writes the completed options information at the end of the card description object (cd.o) file.

B. Reference: Genopt command.

ASKCODES siteid 5 KEY
KEY = 3638, 4393, or 5488
Generic 3:
ASKCODES siteid KEY CARD
KEY = 5, e ▶(PDP-11/70 only), or f◀
CARD = 3638, 4393, 5488, or ▶xotf◀

A. Description: The askcodes command is an interactive routine which must be run for No. 5 crossbar offices. For Generic 3, this routine must be run for No. 5 ETS crossbar offices ▶(PDP-11/70 only) and

No. 5 office test frame (OTF) offices.♦ There are certain fields in some card descriptions where coding types may vary from office to office; for example, the CS field on a 4393 card may be 1/30, 2/30, or 3/30. The askcode program reads the /atadata/fldcodes.xb? file to find out which fields are involved and what the coding alternatives for each field are. It then lists the fields and asks for yes or no decisions from the operator. The results are written to the fldcodes file in the office directory and used to modify the coding in the card description.

B. Example: **ASKCODES TRAIN5 5 4393**

System response:

```
FIELD: CS, CODING ALTERNATIVES ARE:
1/n 2/30 3/30
ENTER CODE - <1/n>

FIELD:  CT, CODING ALTERNATIVES ARE:
1/n 2/5
ENTER CODE - <1/n>

FIELD: FT, CODING ALTERNATIVES ARE:
1/n 2/4
ENTER CODE - <2/4>

FIELD: FT', CODING ALTERNATIVES ARE:
1/n 2/4
ENTER CODE - <2/4>

FIELD CODING SPECIFICATION COMPLETE
```

Generic 3:

System response:

```
FIELD: CS, CODING ALTERNATIVES ARE:
1/n 2/30 3/30
ENTER CODE - <1/n>

FIELD: CT, CODING ALTERNATIVES ARE:
1/n 2/5
ENTER CODE - <1/n>

FIELD: FT&FT', CODING ALTERNATIVES ARE:
1/n 2/4
ENTER CODE - <2/4>

FIELD CODING SPECIFICATION COMPLETE
```

♦Generic 3:
ATA

A. Description: The ATA command changes the switching control center (SCC) terminals that have both modes from UNIX operating system mode to the ATA mode. This command determines the line number involved with the request, checks that the line is associated with an ATA site, and initiates a line control to restore ATA for the desired site.♦

♦*B.* ***Examples:***

1. A change is initiated.

```
ATA
```

System response:

```
ATA MODE CHANGE INITIATED

OK

%
```

2. A change is denied.

```
ATA
```

System response:

```
PERMISSION DENIED, ATA NOT CURRENTLY RUNNING
```

C. ***References:*** Addmon and UNIX operating system commands.♦

BACKUP [EPOCH, SAVEI, GETI]

A. Description: The backup command performs either an epoch, incremental backup, or an incremental restore.

If the keyword epoch is used, the backup program will attempt to copy all of the ATA file system residing on the moving-head disk onto magnetic tape. Each backup is normally taken when the system is operating at run level 6.

The backup program executes **/ETC/DUMPON1** in order to perform the actual backup. If the "dumponl" program terminates abnormally, the backup program will notify the user as to why it did so. If the dumponl program terminates normally, the backup program will update the date in the /etc/epoch file to the date at which the backup was initiated. This date is used by the incremental backup.

If the keyword savei is used, the backup program will look through the entire file system and output onto tape any file which has changed since the last epoch backup was taken. Incremental backups may be taken at any time; however, it is recommended that they be taken when system access is at a minimum.

The backup program executes **/ETC/CMPFS -ETR** /,♦**/ETC/CMPFS -0ERS** / (Generic 3, PDP-11/40),♦ **/ETC/CMPFS -8ERS** / (Generic 3, PDP-11/70) to perform the actual backup. If the cmpfs program terminates abnormally, the backup program will notify the user. In no case will the date /etc/epoch file be changed when taking an incremental backup. As a result, incremental backups are cumulative; ie, each incremental backup supersedes the immediate preceding one. Since an epoch backup backs up all data, taking an epoch backup makes all previous incremental tapes obsolete (except as possible long-term backup).

If the keyword geti is used, the backup program will load the contents of the incremental backup tape into the system. ♦The backup program executes **/ETC/CMPFS -ETR/**, **/ETC/CMPFS -0X** / (Generic 3, PDP-11/40), or **/ETC/CMPFS -8X** / (Generic 3, PDP-11/70) to perform the actual backup.♦

B. Examples:

1. The keyword typed was neither epoch, savei, nor geti.

   System response:
   INVALID DUMP TYPE

2. The program name could not be found in the file system. This is usually an indication that the disk has been scribbled.

   System response:
   CANNOT EXECUTE (name)

3. The program could not create the epoch date file. This probably means that either the disk has been scribbled or that the backup program does not have the correct privileges to create the file.

   System response:
   CANNOT CREATE /ETC/EPOCH

4. The subprogram which backup executes dies abnormally.

   System response:
   "/ETC/DUMPON1" FATAL ERROR n
   The number n is the error number. The only case which should ever occur would be the result of the user typing **RUBOUT** at the terminal in order to abort the backup. In this case, the system response should be: **"/ETC/DUMPON1" FATAL ERROR 2**

Generic 3 ♦with PDP-11/70:♦
BACKUPMON

A. Description: The backupmon command allows certain ATA sites to monitor for a message indicating that an ETS dial backup line has become active. The backupmon command builds a table that is a list of sites that are to receive the announcement that ETS is on a backup line. A transfer command must be made to the backup line before the connection is dropped by ETS.

B. Example: Monitor for a message indicating that an ETS dial backup line has become active.

BACKUPMON

System response:

```
ADD RMV PRT EXIT
<ADD>


SITE NAME TO RECEIVE BACKUP MESSAGE - <SCC>
ADD RMV PRT EXIT
<ADD>


SITE NAME TO RECEIVE BACKUP MESSAGE - <PING>
ADD RMV PRT EXIT
<ADD>
```

```
SITE NAME TO RECEIVE BACKUP MESSAGE - <AOS>
ADD RMV PRT EXIT
<ADD>

SITE NAME TO RECEIVE BACKUP MESSAGE - <SCC>
SCC ALREADY ON BACKUP LIST

ADD RMV PRT EXIT
<PRT>
SITE SCC LOOKING FOR BACKUP MESSAGE
SITE PING LOOKING FOR BACKUP MESSAGE
SITE AOS LOOKING FOR BACKUP MESSAGE

ADD RMV PRT EXIT
<RMV>
SITE SCC LOOKING FOR BACKUP MESSAGE
SITE AOS LOOKING FOR BACKUP MESSAGE

ADD RMV PRT EXIT
<EXIT>
```

Generic 3:
BROADCAST "text"

A. Description: The broadcast command causes the printing of the text message at all ATA sites and UNIX operating system terminals.

Generic 3 ♦with PDP-11/70:♦
BROWSE
BROWSE FILENAME
B
B FILENAME

A. Description: ♦The browse command is only used with the VT100 terminal.♦ To use the browse command, login as history. This login gives the user restricted UNIX operating system capabilities. The browse command allows the filename file to be displayed on a CRT work station. When called, it fills the screen with the first set of messages. The display is then controlled with the following keys:

KEY	FUNCTION
PF1	Causes the messages to scroll up. Subsequent depression will stop scrolling.
PF2	Causes the messages to scroll down.
PF3	Causes messages to scroll up one message and stop.
PF4	Causes messages to scroll down one message and stop.
n	Causes the screen to jump to a previously marked position in the file. If that digit (1 through 9) has not been used in a mark command, then no jump takes place.
0	Returns a file to a position after a jump.

RETURN	Puts the browser in the command mode.
—	Erases the last character typed (when in the command mode).
$	Erases all characters typed on the line (when in the command mode).
NO SCROLL	Stops scrolling. Subsequent depression of this key starts scrolling.

Caution: When this key is used to stop scrolling, no other key will function until scrolling is restarted.

DELETE	Terminates browsing and returns control to the UNIX operating system command mode.

Alphabetic characters (a through z) place the browser in the command mode. Any key not described above is ignored and scrolling stops. If no file name is specified, the program restarts using data saved when the browser was last executed. File name should always be entered when first logging on the system.

When in the command mode, the browser accepts command in the same manner as the ATA command mode. A valid list of commands is given in Table B.

B. Reference: Convert command.

CAT [-u] FILE...

A. Description: The concatenate command reads each file in sequence and writes it on the standard output. Thus, **CAT FILE** prints the file and **CAT FILE1 FILE2> FILE3** links the first two files and places the result on the third. If the optional **-u** argument is used, a write occurs every wakeup as opposed to every 512 bytes. If no input file is given, or if the argument "-" is encountered, **CAT** reads from the standard input file.

B. Reference: Copy (cp) command.

CHDIR DIRECTORY
CD DIRECTORY

A. Description: Since a new process is created to execute each command, **CHDIR** or **CD** would be ineffective if written as a normal command; therefore, it is recognized and executed by the shell program.

DIRECTORY becomes the new working directory. The process must have execute (search) permission in **DIRECTORY**.

B. Reference: Shell (sh) command.

CHECK [-SN][file system]

Note: Do not use check unless system is in run level 6 or 7.

A. Description: The check command audits UNIX operating system files for consistency and corrects any discrepancies. Since these corrections will, in general, result in a loss of data, the program will

TABLE B

BROWSE COMMANDS

COMMAND	DESCRIPTION
save! save:fname!	This command appends the message at the top of the screen to the save file. If no file name is given, the last file name used is understood. Upon initialization, this name is set to the save file. If the message at the top of the screen has an expansion associated with it, then the expansion is also saved.
mark:n! n=1 thru 9	This command marks the current position of the screen for subsequent jumps. If the current screen contains an expansion, then only the original message is remembered.
expand e!	This command expands the message at the top of the screen into a more readable form. The screen is cleared and the expansion is displayed. It may be examined by scrolling like any other message. As soon as both the original message and the expansion are off the screen, the expansion is off. Typically, output messages from No. 5 ETS are compact. The expand program reformats complicated messages into a more usable form.
browse:pat (pattern)! b:pat (pattern)	This command causes the browser to display only those messages which fit the pattern.
browse:fname! browse! b:fname! b!	This command is identical to the UNIX command. The current browser is terminated and a new program is executed using the specific file name. If the input file name cannot be opened, the browser terminates and returns to the UNIX command mode.
clear!	This command removes the effect on the browse:pat! command. The clear command should be performed before attempting another pattern command, such as bkwd or fwd.
fwd:pat (pattern)! bkwd:pat (pattern)!	This command causes the browser to search forward (or backward) from the current position. If a match is found, the browser jumps to the matched message. If no match can be found, an error message is printed.
jump:current! jump:eof! jump:sof! jump:(time)! j:current! j:eof! j:sof! j:(time)!	This command causes the browser to jump to the specified position.
convert:key data! con:key data!	This command performs data conversions and translations.

request operator concurrence for each such action. All questions should be answered by typing yes or no, followed by a line feed. Typing yes will cause the correction to take place. However, if the program does not have write permission on the file system or the no option (-N) is on, then all questions will automatically be answered no.

The program consists of six separate phases. Some phases are skipped if they are not needed. In phase one, check examines all block pointers in all files, checking for pointers which are outside the file system (BAD) and for blocks which appear in more than one file (DUP). A table is made of all DUP blocks and all defective files are marked for clearing. Each error is printed, but no correction takes place in this phase.

The second phase is run only if DUP blocks were found in phase one. This phase finds the rest of the DUP blocks, marking each for clearing.

The third phase now checks the directory structure of the file system. This is done by descending the directory tree, examining each entry. A count is kept of the number of references to each file. If an entry refers to an unallocated file, a file marked for clearing, or a file number outside the file system, then the entry is printed. If the operator agrees, it is removed. Refusing to remove an entry to a marked file will clear the mark, preserving the file and its subsequent entries.

In phase four, all marked or unreferenced files are listed. With concurrence from the operator, each of these files is then cleared. In addition, any file whose link count does not agree with the number of references is listed, and, if agreed, the link count is adjusted.

If the salvage option (-S) is on, then phase five is skipped. Otherwise, check examines the free list. If any blocks are found which are outside the file system or which have been previously encountered in a file or elsewhere in the free list, then the list is pronounced BAD and a salvage is called for. Agreement will set the salvage option and proceed to the next phase. If there are no defects in the free list and all blocks are accounted for, the check is finished. Otherwise, the number of missing blocks is printed and a salvage is requested.

The last phase is the salvage operation, where the free list is recreated. It is run whenever the salvage option (-S) is on, or a problem has been found with the free list. Simply stated, a new free list is constructed containing all blocks not found in some file.

B. Examples:

The system responses are in general self-explanatory. In the examples that follow, these notations will be used:

<b>	block number
<i>	inode number
<fname>	file pathname
<n>	positive integer
<c>	option character.

Check begins with the following output:

```
<filesystem>[(NO WRITE)]
```

```
Phase 1
```

The no-write message indicates that the program does not have write permission on the file system. Therefore, subsequent corrections will be suppressed by automatically answering no to all questions.

Phase one then proceeds to list any BAD or DUP blocks and their inode number, as follows:

```
<b>    BAD                I = <i>
<b>    DUP                I = <i>
<b>    EXCESSIVE DUPS     I = <i>
```

If too many DUPs are encountered, the program will list all blocks, but it will not mark the excess DUPs for later processing. When phase 1 is finished, if any DUPs were encountered, then phase 2 is run. Otherwise, phase 2 is skipped. This phase will list the rest of the DUP blocks as follows:

Phase 2

```
<b> DUP I = <i>
```

Check now descends the directory tree, asking to remove any defective entries.

Phase 3

```
I OUT OF RANGE    I = <i>   <fname>   REMOVE?
UNALLOCATED       I = <i>   <fname>   REMOVE?
BAD/DUP           I = <i>   <fname>   REMOVE?
```

Unless the no option is on, the program will wait for a response of yes or no after each question. A no answer to the BAD/DUP entry will unmark that inode for clearing. This will suppress any subsequent correction to that file.

Now, check will clear or adjust any defective files. Again, it will wait for a yes or no response to each question. The program will also indicate whether each entry is a file or a directory.

Phase 4

```
UNREFERENCED   [FILE/DIRECTORY]   I = <i>   CLEAR?
BAD/DUP        [FILE/DIRECTORY]   I = <i>   CLEAR?
LINK COUNT     [FILE/DIRECTORY]   I = <i>   ADJUST?
```

If the salvage option (-S) is not on, the program will now validate the free list. Otherwise, this phase is skipped. If there are any errors in the free list, it will specify them and request a salvage.

Phase 5

```
BAD FREE LIST                    SALVAGE?
<n> MISSING                      SALVAGE?
```

Phase 6 is the salvage operation. It is only done if one has been requested.

Phase 6

Finally, some totals are printed: the total number of allocated files (including directories and special files), the number of blocks in use, and the number of blocks in the free list.

```
<n> FILES <n> BLOCKS <n> FREE
```

If the file system has been modified, then the following message is printed and the program goes into a loop. This is only a reminder to the operator since the program can be forced to terminate with a <DEL> character. ▶If the filesystem is the root (/dev/rootdev) filesystem, then the system must be rebooted. The DEL will not terminate the program.◀

```
******BOOT UNIX(NO SYNC!)*****
```

While running, a number of errors can occur which cause the program to terminate. An illegal option or the inability to open the filesystem are shown as:

```
<c> OPTION?
CAN NOT OPEN <filesystem>
```

An I/O error on the filesystem will also cause an error message. In this case, the operator is given the choice of exiting (yes) or continuing (no). This error is generally a hardware error, and continuing is rarely a good idea.

```
CAN NOT READ   <filesystem>    BLOCK <b>    EXIT?
CAN NOT SEEK   <filesystem>    BLOCK <b>    EXIT?
CAN NOT WRITE  <filesystem>    BLOCK <b>    EXIT?
```

▶Generic 3 with PDP-11/70:
CHGHIST sitename

A. ***Description:*** The change history command either increases or decreases the size of the ETS history logging file. History logging must be created with the addoffice program before this program can be used. The argument sitename is required.

B. ***Example:*** **CHGHIST sitename**

System response:

```
DEDICATED BLOCKS:  49742, USED BLOCKS:  600
AVAILABLE BLOCKS:  49142

ENTER NUMBER OF BLOCKS:  <1000>
```

C. ***References:*** Addoffice and rmsite commands.◀

COMPILE siteid KEY
KEY = 1 or 5
Generic 3:
COMPILE siteid KEY CARD
KEY = 1, 5, e ▶(PDP-11/70 only), or f◀
CARD = 3638, 4393, 5488, tind, or ▶xotf◀

A. ***Description:*** The compile command creates the analysis I tree binary file in anal1.0 under the office directory. There are several steps to this process. The analysis I master file for the crossbar type is read from /atadata/a1.master.xb5. The prime program uses the features file to remove the unneeded sections from the master file, producing anal1.pruned. The CSL prepass program reads this file, removes useless labels from the CSL source code, and writes the trimmed file to anal1.src. Finally, passes 0, 1, and 2 of the CSL compiler are run, producing the final binary tree in "anal1.0".

If a mistake is found in the yesno answer, **YESNO** and **COMPILE** must be rerun for the office.

If this office's anall.0 file was copied to any other offices as a shortcut, the new anall.0 and features files should be copied to the same offices.

LOADRUN should be rerun to correct runtime data.

B. Reference: Yesno command.

Generic 3 ▶with PDP-11/70:◀
CONVERT KEY DATA
CONV KEY DATA

A. Description: The convert command performs data conversions and translations. Key identifies which conversion is to be performed. The following is a list of available conversions:

KEY	**FUNCTION**
oct	Octal to decimal
dec	Decimal to octal
o.hex	Octal to hexadecimal
h.oct	Hexadecimal to octal
bs0	Decimal to binary and octal
bs2	Binary to decimal and octal
bs8	Octal to binary and decimal
calc0	Simple calculator for decimal numbers
calc2	Simple calculator for binary numbers
calc8	Simple calculator for octal numbers
calc16	Simple calculator for hexadecimal numbers
list	List options.

The data keyword identifies the data that is to be converted. This field is not required for the key field calc0, calc2, calc8, calc16, and list.

B. Examples:

1. Convert octal number 1076 to decimal.

```
CON OCT 1076
```

System response:

```
OK DECIMAL = 574
```

2. Convert hexadecimal number abcd to octal.

```
CONV HEX.O ABCD
```

System response:
OK OCTAL - 2536336

CP [-t] <fname> <ufname>

A. Description: The copy command copies the contents of the fname file into a new ufname file. The mode and owner of the target file are preserved if it existed; the mode of the source file is used otherwise.

If ufname is a directory, then the target file is a file in the directory with a file name of fname.

The [-t] option causes CP to tell how may writes it has performed.

B. Example: **CP /TMP/EL EL.SAVE**

System response: The /tmp/el file is copied into an el.save file residing in the user's current directory. If /tmp/el file does not exist, response will be **CANNOT OPEN OLD FILE**; if no write permission for target file, response will be **CAN'T CREATE NEW FILE.**

DATE [mmddhhnn] [yy]

A. Description: If the optional argument is used, this command enters the given date and time into the ATA internal clock system. The argument must contain eight numeric characters; however, it can contain ten characters, with the last two being the year.

mm = month (01-12)

dd = day (01-31)

hh = hour (00-23)

nn = minute (00-59)

yy = year (70-98).

If the optional argument is not used, this command outputs the system's idea of the current time in the following format:

day mmm dd hh:mm:ss yr

day = day of the week

mmm = month

dd = day

hh = hours

mm = minutes

ss = seconds

yr = year.

B. Example: Obtain the current date.

```
DATE
```

System response:

```
SAT SEP 27 09:46:45 1976
```

If the optional argument is used and a nonnumeric character is entered in the time field or less than eight characters are used, the system response will be **BAD CONVERSION**.

Generic 3 ▶with PDP-11/70:◀
DMPEM TC FILE

A. Description: This command is used to dump an error message from an ETS type site, displaying the error message type and number and the arguments and contents of the 16 registers.

B. Example: Dump an error message from an ETS type site.

```
DMPEM /OFFICE/TC/TC07
```

System response:

```
54 BYTES OF CARD DATA READ
SEQUENCE NUMBER = 0 IN DECIMAL; 0 IN OCTAL
```

Error message data is printed out.

C. Reference: Lograw command.

Generic 3 ▶with PDP-11/70:◀
DMPETS TC FILE

A. Description: This command is used to dump a card image from an ETS type site in a format similar to a trouble card.

B. Example: Dump a card image from an ETS type site.

```
DMPETS /OFFICE/TC/TC03
```

System response:

```
192 BYTES OF CARD DATA READ
SEQUENCE NUMBER = 0 IN DECIMAL; 0 IN OCTAL
```

The card image is printed out.

C. Reference: Lograw command.

Generic 3:
DMPMDT TC FILE

A. Description: This command is used to dump a card image from an MDT type site in a format similar to a trouble card.

B. Example: Dump a card image from an MDT type site.

```
DMPMDT /OFFICE/TC/TC03
```

System response:
192 BYTES READ

The card image is printed out.

C. Reference: Lograw command.

Generic 3:
DMPOTF TC FILE

A. Description: This command is used to dump an indicator image from an OTF type site in a format similar to a trouble indicator display.

B. Reference: Lograw command.

Generic 3:
DMPTPIU TC FILE

A. Description: This command is used to dump a card image from a TPIU type site in a format similar to a trouble card.

B. Example: Dump a card image from a TPIU type site.

```
DMPTPIU /OFFICE/TC/TC11
```

System response:
192 BYTES READ

The card image is printed out.

C. Reference: Lograw command.

GENOPT sitename KEY1 KEY2
KEY1 = 1 or 5
KEY2 = 3638, 4393, 5488, or tind
Generic 3:
GENOPT sitename KEY1 CARD
KEY1 = 1, 5, e ♦(PDP-11/70 only), or f♦
CARD = 3638, 4393, 5488, tind, or ♦xotf♦

A. Description: The genopt command is an interactive routine. Genopt is used to generate the data referred to as the office options, which are needed to supplement the information found in the card description and the trouble record in order to properly bin troubles. Examples of office options might be the number of markers which are the wire spring type, or the called numbers which are being screened for annoyance calls.

The program prints a message describing the information needed. The operator responds by entering a list of numbers separated by blanks and terminated by a carriage return. The list represents appropriate equipment numbers as requested by the prompting message.

In case of erroneous or unrecognizable operator input, the program prints the values which were rejected and requests that they be reentered. Any inputted numbers which are not specifically printed as rejected were accepted and should not be reentered along with the bad inputs. If no numbers should be entered to replace the rejected ones, a carriage return causes the program to proceed to the next option.

The answers to the office option questions are used to create the data which is appended to the card description object file "cd.o" under the office directory by the addopt program.

B. Example: **GENOPT TRAIN5 5 4393**

System response:

```
INSTRUCTIONS?  (Y-N) - <Y>

WHEN PROMPTED WITH '-', ENTER THE APPROPRIATE NUMBERS, SEPARATED BY BLANKS. CARRIAGE
RETURN TERMINATES INPUT. IF THERE ARE TOO MANY NUMBERS TO FIT ON A LINE, FOLLOW
THE LAST NUMBER ENTERED WITH A BLANK, AN AMPERSAND ("&"), AND A CARRIAGE RETURN.
THIS WILL CAUSE YOU TO BE PROMPTED FOR FURTHER INPUT.

ENTER THE MARKER NUMBERS (0-19) OF ALL WIRE-SPRING MARKERS -
<0 1 2 3 4 5 6 7 8 9 10 11>

ENTER THE MARKER NUMBERS (0-19) OF ALL DIAL TONE MARKERS -
<3 4 5 8>

ENTER 3-DIGIT OFFICE CODES FOR AUTOMATIC NUMBER
ANNOUNCEMENT OR RINGBACK - <958>

ENTER NUMBERS OF ALL (1000-TYPE) AMA TRANSLATORS -
<0 1 2 3 4 5 6 7 8>

ENTER THE NUMBERS OF ALL CAMA TRANSVERTERS -
<CR>

ENTER THE NUMBERS OF ALL ANI TRANSVERTERS (SD-26161-01) -
<CR>
```

LCTRL siteid KEY OPTION
KEY = RESTORE, REMOVE, or TRANSFER LNXX, LNYY
OPTION = ATA, TTY, QUEUE, or ALL

A. Description: The line control command is used to restore, remove, or transfer lines. The restore option enables the operation of a line associated with a site as it pertains to the various input/output functions of the ATA System. The ATA keyword enables the line for trouble input logging, the TTY keyword enables the line to be the source of ATA commands, and the QUEUE keyword enables the line to receive output from the ATA System.

The remove option removes a line connected to a site from logger, shell, or queue. Removal of a line from ATA disconnects it from the logger program which monitors for trouble data input from a site. Removal from the shell program disconnects it from the ATA shell which monitors for commands from the line. Removal from the queue disconnects it from the output processing programs of the ATA System.

The transfer option transfers line data from one line (LNXX) to another (LNYY).

B. Examples:

1. Restore the line associated with site Clinton to the monitoring of trouble inputs.

```
LCTRL CLINTON RESTORE ATA
```

System response:

```
CLINTON:
     LOGGING IS ON
```

2. Remove the line connected with site Clinton from the program which monitors for input trouble card or indicator data.

```
LCTRL CLINTON REMOVE ATA
```

System response:

```
CLINTON:
     LOGGING IS OFF
```

3. Transfer line 50 to line 60.

```
LCTRL CLINTON TRANSFER LN50 LN60
```

System response:

```
SITE CLINTON NOW HAS LINE 60
     LOGGING IS ON
     QUEUE IS ON
     TTY IS ON
```

C. References: Remove, restore, and transfer commands in Section 190-105-301.

Generic 3:
LCTRL siteid KEY OPTION
KEY = RESTORE, REMOVE, or TRANSFER LNXX, LNYY
OPTION = ATA, KB, HISTORY ♦(PDP-11/70 only),♦ UNIX operating system, or ALL

A. Description: The line control command is used to restore, remove, or transfer lines. The restore option enables the operation of a line associated with a site as it pertains to the various input/output functions of the ATA System or the UNIX operating system. The ATA keyword enables the line for trouble input logging; the KB keyword enables the line to be the source of ATA commands and to receive output from the ATA System. The UNIX operating system keyword enables the line to be the source of the UNIX operating system commands. When the line is in the UNIX operating system mode, its

KB functions are disabled. When the line is in the ATA mode, its UNIX operating system functions are disabled. The HISTORY keyword enables history logging for a No. 5 Crossbar ETS site.

The remove option removes a line connected to a site from the logger, the shell and queue, the history logger, or the UNIX operating system. The ATA, KB, HISTORY, and UNIX operating system keywords disable the line for those functions described under the restore option. For work stations, if UNIX operating system is disabled, the KB functions are automatically enabled for that line.

The transfer option transfers line data from one line to another. If a work station is working in the UNIX operating system mode when it is transferred, it operates from the new line in the UNIX operating system mode. If it was operating in the KB mode, it operates from the new line in the KB mode. When the site is transferred, its KB, ATA, and HISTORY options are restored.

B. Examples:

1. Restore the line associated with site Clinton to process trouble inputs.

```
LCTRL CLINTON RESTORE ATA
```

System response:

```
CLINTON:
     LOGGING IS ON
     KEYBOARD IS *
```

Note: An asterisk indicates the current state (on or off).

2. Remove the line connected with site Clinton from the program which monitors for input trouble card or indicator data.

```
LCTRL CLINTON REMOVE ATA
```

System response:

```
CLINTON:
     LOGGING IS OFF
     KEYBOARD IS *
```

3. Transfer line 50 to line 60.

```
LCTRL CLINTON TRANSFER LN50 LN60
```

System response:

```
ACTIVATING CHANGES NOW
SITE CLINTON NOW HAS LINE 60

CLINTON:
     LOGGING IS ON
     KEYBOARD IS ON
```

LDCD sitename
Generic 3:
LDCD sitename KEY CARD
KEY = 1, 5, e ▶(PDP-11/70 only), or f◀
CARD = 3638, 4393, 5488, tind, or ▶xotf◀

A. ***Description:*** The load card command uses the cd.src and the fldcodes files to build the cd.o file under the office directory. No interaction is required.

If a mistake is made in the askcodes routine, then askcodes, ldcd, and addopt are rerun for the office, and then loadrun is run again to change the runtime data base.

B. ***References:*** Askcodes and addopt commands.

LOADRUN KEY
KEY = 1 or 5
Generic 3:
KEY = 1, 5, e ◗(PDP-11/70 only), or f◖

A. ***Description:*** The loadrun command is a shell program that calls the bsetup and iahsetup programs. It copies the standard office sorts, bin names, and bin number files into the runtime data base.

The bsetup program builds the runtime binary tree and card description file for all of the offices in /run.dir/a1.bcode. This file is built from the cd.o and anal1.0 files in the office directories. A header file containing offsets and lengths for the information for each site is created in /run.dir/a1.bhdr. No interaction is required.

The iahsetup program builds the runtime IA header file for all of the offices in /run.dir/ia.hdr. The file is built from the iah.o files in each of the office directories. No interaction is required.

B. ***Example:*** **LOADRUN 5**

System response:

```
RUNTIME DATA LOADING BEGINS

BSETUP CALLED FOR:

JFKO    SITE 0
CENO    SITE 1
RSN1    SITE 2

IAHSETUP CALLED FOR:

JFKO    SITE 0
CENO    SITE 1
RSN1    SITE 2
```

LOGRAW [site/line] [name/number] KEY
site/line = keywords
name = site name
number = line number
KEY = yes or no
Generic 3:
KEY = on or off

A. ***Description:*** The lograw command sets or resets the flags in the line translation table to cause raw card images to be dumped by the logger. The raw images are stored in the /office/tc directory. The file name has the form tc00 through tc24. Each line or site that has been flagged for lograw will have each incoming message placed in one of the tc files. The files will be overwritten when full. Raw dumping may be specified by name or number.

B. Example: **LOGRAW SITE JFKO ON**

System response:
OK

C. Reference: Section 190-105-312.

LOOP lnxx lnyy [speed]
speed = 300 or 1200
Generic 3:
LOOP lnxx lnyy [baud rate]
baud rate = 300, 1200, ▶1800, or 2400◀

Note: The speed and baud rate options are used only with the PDP-11/70.

A. Description: The loop command checks the ability of the system to send a standard message over line xx and receive it over line yy. The loop command executes the testsend program on line xx and the testrcv program on line yy. If no speed or baud rate is given, loop assumes 300 baud. The cables for the two lines must be connected using a standard null modem cable (ED-1P123-30, group 1).

B. Example: Enter the loop command. System response will be **ATA LOOP TEST** printed repetitively on the requesting terminal. This will continue until **DEL** or **RUBOUT** is typed on the requesting terminal. The system response will be **TESTRCV PROGRAM DIED** or **TESTSEND PROGRAM DIED** and the loop program exits. If either the testrcv or testsend program fails to die, the loop program will hang. Should this happen, it may be forced to exit by typing another **DEL** or **RUBOUT**. The system response will be **TESTSEND PROGRAM REFUSED TO DIE.**

C. References: Transmit and receive commands in Section 190-105-301.

LS [-1tasdruifl] [filesys] NAME ...

A. Description: For each directory, the ls command lists the contents of the directory. For each file argument, the ls command repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately but file arguments appear before directories and contents. These are several options:

OPTION	DESCRIPTION
-1	List in long format—given mode, number of links, owner, group, size in bytes, and time of last modification for each file.
-t	Sort by time modified (latest first).
-a	List all entries; usually those beginning with '.' are suppressed.
-s	Give size in blocks for each entry.

-d If argument is a directory, list its name (mostly used with -1 to get status on directory).

-r Reverse the order of sort to get reverse alphabetic or oldest first as appropriate.

-u Use time of last access instead of last modification for sorting (-t) or printing (-1)

-i Print i-number in first column of the report for each file listed.

-f Force each argument to be interpreted as a directory and list the name found in each slot.

-I Causes the next argument to be used as a device containing a UNIX operating system file and interprets the name arguments as an inode number.

MAKEFH (not used in Generic 3)

A. Description: The makefh command is used to build the complete file system on the fixed-head disk. This should be done after a fixed-head disk crash or after Digital Equipment Corporation has run diagnostics.

B. Example: **MAKEFH**

System response:

```
MAKING FIXED HEAD DISK FILE SYSTEM
FIXED HEAD DISK FILE SYSTEM COMPLETED
COPYING FILES FROM MOVING HEAD TO FIXED HEAD DISK
ALL FILES COPIED TO FIXED HEAD DISK
```

/ETC/MKFS SPECIAL PROTO [-b[NUMBERS]]

A. Description: The mkfs command constructs a file system by writing on the special file according to the directions found in the proto file. The proto file contains tokens separated by spaces or new lines. The first token is the name of a file to be copied onto block zero as the bootstrap program. The second token is a number specifying the size of the created file system. Typically, it will be the number of blocks on the device, perhaps diminished by space for swapping. The next token is the i-list size in blocks. An optional third token is the keyword badblocks followed by a list of numbers (decimal) and terminated within the token $. The list specifies blocks that are to be left out of the file system. The next set of tokens consists of the specifications for the root file. File specifications consist of tokens giving the mode, the user-ID, group-ID, and the initial contents of the file. The syntax of the contents field depends on the mode.

The -b option allows specification of badblock numbers on the command line. This may be used in combination with the keyword badblocks in the proto file.

B. Example: **/ETC/MKFS /DEV/ATADEV PROTO.ATA.INIT**

/ETC/MKNOD /DEV/LNXX C [4 6] XX

A. Description: The mknod command is used to restore line entries in the /DEV directory after they have been removed (intentionally or otherwise). The 2-digit line number to be rebuilt into the file system access structure is indicated by XX. The type of line on the system is indicated by 4 (dhs) or 6 (djs). This command can only be executed by root.

B. ***Example:*** **/ETC/MKNOD LN32 C 6 32**

/ETC/MOUNT SPECIAL FILE [-r]

A. ***Description:*** The mount command announces to the system that a removable file system is present on the device corresponding to the special file. The file must exist already; it becomes the name of the root of the newly mounted file system.

The mount program maintains a table of mounted devices; if invoked without an argument, it prints the table.

The -r option indicates that the file is to be mounted read-only. Physically write-protected and magnetic tape file system must be mounted in this way or errors occur when access times are updated.

B. ***Examples:***

1. A removable file system is mounted on the moving head disk.

```
/ETC/MOUNT /DEV/ATADEV /ATA
```

2. To verify that a file system is mounted.

```
/ETC/MOUNT
```

System response:

```
ATADEV ON /ATA
ATASRC ON /USR/EMATA
HISTORY ON /HISTORY
```

▶Generic 3:

3. To verify that a file system is mounted.

```
/ETC/MOUNT
```

System response:

```
ATADEV ON /ATA
RX1 ON /LMDATA◀
```

C. ***Reference:*** Unmount command.

MV <fname1> <fname2>

A. ***Description:*** The move command takes the fname1 file and puts it in the fname2 file. Both fname1 and any file previously called fname2 are destroyed.

B. ***Example:*** **MV /TMP/EL EL.SAVE**

System response: The /tmp/el file is moved to the el.save file residing in the user's current directory. If the source file does not exist, the system response will be **fname1 DOES NOT EXIST.**

OFTYPE NAME KEY CARD
NAME = office name
KEY = 1 or 5
CARD = 3638, 4393, or 5488
Generic 3:
OFTYPE NAME KEY CARD (generic/issue)
NAME = office name
KEY = 1, 5, e ▶(PDP-11/70 only), or f◀
CARD = 3638, 4393, 5488, tind, or ▶xotf◀
generic/issue ▶(PDP-11/70 only)◀ = No. 5 ETS generic and issue in the form x.y

A. Description: The oftype command is a shell program that calls trblink, ldsgt, ldcd, addopt, compile, ldiah, and ldsite programs.

B. Example: **OFTYPE TRAIN5 5 4393**

System response:

```
BEGINNING TO PROCESS OFFICE TRAIN5
TOTAL SORT GROUPS PROCESSED = XXX
OFFICE DATA BEGINS AT WORD XXXX
FIELD CODING MODIFICATION COMPLETED
CARD DESCRIPTION TRANSLATION COMPLETED
OPTIONS ADDITION COMPLETED
COMPILING /OFFICE/TRAIN5
TREE PRUNING COMPLETED
PREPASS COMPLETED

PASS 0

READING XXX FIELD NAMES
LAST FIELD NAME WAS:  N'7

READING BIN NAMES
XXX BIN NAMES; LAST NAME WAS:  XWB

PASS 0 COMPLETED

PASS I

READING TERMINATED:  READ=0
LINE COUNT = XXXX, LOCATION COUNTER = XXXX (OCTAL BYTES)

PASS I COMPLETED

PASS II

PASS II COMPLETED
IA HEADER LOADING COMPLETED
```

```
RTDATA BUILD COMPLETED FOR TRAIN5
OFTYPE PROCESSING COMPLETED FOR TRAIN5
```

Generic 3 ▶with PDP-11/70:◀
OPPAT PNAME

A. Description: The oppat command outputs the definition for the pattern specified by pname. Patterns are divided into two categories: user defined and common. In user defined patterns, the user's directory (/usr/history) is searched for a file named pname.p. In other patterns, the common pattern directory (/compat) is searched. To use this command, login as history. This login gives the user restricted UNIX operating system capabilities.

B. Examples:

1. Definition of a user pattern.

```
OPPAT DAS
```

System response:

```
DAS:      (USER DEFINED)

SOMTAB TAB(2) F(3) "DAS"

EXPANDED OUTPUT?: <Y>

EXPANDED PATTERN OUTPUT:

SOMTAB TAB(2) FIELD(3) "DAS"
```

2. Definition of a common pattern.

```
OPPAT TAB
```

System response:

```
TAB:

TAB

TAB OR T    THIS SKIPS THE NEXT CHARACTER.
           (TAB IS EQUIVALENT TO ALL.)
            TAB MAY BE ABBREVIATED T.

TAB(N) OR T(N)    THIS SKIPS THE NEXT N CHARACTERS.
                  TAB(0) ALWAYS FAILS. TAB(N) MAY
                  BE ABBREVIATED AS T(N).

EXPANDED OUTPUT:  <N>
```

C. Reference: Patgen command.

PASSWD NAME PASSWORD

A. Description: The password command is placed on the given login name. This can only be done by the person corresponding to the login name or by the superuser. An explicit null argument (#) for the password argument removes any password from the login name.

B. Example: **PASSWD ROOT WHOGAS**

Generic 3 ▶with PDP-11/70:◀

PATGEN PNAME

A. Description: The patgen command enables the user to define a pattern for use with the browser command. The pattern is stored in the user's directory in the pname.p file. To make patterns, login as history and run the patgen program. This login gives the user restricted UNIX operating system capabilities.

B. Example: **PATGEN DAS**

System response:

```
DATA: SOMTAB TAB(2) F(3) "DAS" <CR>
DATA:  CR
OK
[SOMTAB TAB(2) F(3) "DAS"]
```

C. Reference: Oppat command.

REMOVE siteid.KEY (not used in Generic 3)

KEY = ATA, TTY, or QUEUE

A. Description: The remove command removes a line connected to a site from logger, shell, or queue. Removal of a line from ATA disconnects it from the logger program which monitors for trouble data input from a site. Removal from the shell disconnects it from the ATA shell which monitors for commands from the line. Removal from the queue disconnects it from the output processing programs of the ATA System.

B. Examples:

1. Remove the line connected with site Clinton from the program which monitors for input trouble card or indicator data.

```
REMOVE CLINTON.ATA
```

System response:

```
OK (LOGGING IS OFF)
```

2. Remove the line associated with site Clinton from the programs monitoring for command input from that site.

```
REMOVE CLINTON.TTY
```

System response:

```
OK (TTY IS INACTIVE)
```

3. Remove the line associated with site Clinton from the programs which process output from the ATA System.

   REMOVE CLINTON.QUEUE

   System response:
   OK (QUEUE INACTIVE)

C. References: Restore, transfer, and line control commands in Section 190-105-301.

RESTORE siteid.KEY (not used in Generic 3)
KEY = ATA, TTY, or QUEUE

A. Description: The restore command is used to enable the operation of a line associated with a site as it pertains to the various input/output functions of the ATA System. The ATA keyword enables the line to trouble input logging, the TTY keyword enables the line to be the source of ATA commands, and the QUEUE keyword enables the line to receive output from the ATA System.

B. Examples:

1. Restore the line associated with site Clinton to the monitoring of trouble inputs.

   RESTORE CLINTON.ATA

   System response:
   OK (LOGGING IS ON)

2. Restore the line associated with site Clinton to the programs which receive ATA commands.

   RESTORE CLINTON.TTY

   System response:
   OK (TTY IS ACTIVE)

3. Restore the line associated with site Clinton to output from the ATA System.

   RESTORE CLINTON.QUEUE

   System response:
   OK (QUEUE ACTIVE)

C. References: Remove, transfer, and line control commands in Section 190-105-301.

RM [-f] [-r] [-d] NAME ...

A. Description: The rm command removes the entries for one or more files from a directory. If an entry was the last link to the file, the file is destroyed. A write permission on the file is required to remove that file.

If there is no write permission to a file designated to be removed, the rm program prints the file name, its mode, and then reads a line from standard input.

If a designated file is a directory, an error comment is printed unless either the -r or -d option has been used. When the -r option is used, the rm program recursively deletes the entire contents of the specified directory. When the -d option is used, the rm program tries to remove the entry for the specified directory name.

B. Reference: Rmdir command.

▶Generic 3 with PDP-11/70
RMALMMON

A. Description: The rmalmmon command removes the line assigned to the ATA watchdog monitor.

B. Example: Remove line assigned to watchdog monitor.

```
RMALMMON
```

System response:

```
$
```

Generic 3 with PDP-11/70:◀
RMBACKUP

A. Description: The rmbackup command allows lines to be removed from dial backup status.

B. Example: Remove a line from dial backup status.

```
RMBACKUP
```

System response:

```
LINE 37 ASSIGNED TO ETS DIAL BACKUP

BACKUP LINE NUMBER TO BE REMOVED?  <37>
```

RMDIR [-r] DIRECTORY

A. Description: The rmdir command removes directories. The directory must be empty unless the -r option is used.

When the -r option is used, the rmdir program recursively removes directories. The directories must be empty except for other possible directories. If the name given is '.', the current directory is not removed.

B. Reference: Rm command.

RMSITE   sitename

A. Description: The rmsite command is used to remove a site from an ATA System. The entries which existed for the site are cleared in the sitedata, linedata, line translation, and site name tables. The entries in the message distribution, command acceptance, and command reach matrices concerning the site are cleared as well as these same matrices in any saved configuration. The ERD for the removed site is cleared as well as any records that existed in the MRS for that site. The ER sequence number is reset to zero. If the site is an ETS site, the entries in the history logging control blocks are also removed. There are three steps in the removal of an office from an ATA System: remove the site, remove the subdirectory, and remove the runtime data base.

B. Examples:

1. Remove site 0 (Clinton) from ATA (11/40 system).

```
RMSITE CLINTON
```

System response:

```
CLINTON FOUND

<RM -F /OFFICE/CLINTON/*>

<RMDIR /OFFICE/CLINTON>

<RM -F /RUN.DIR/MOUNT.DIR/RTDATA0>
<RM -F /RUN.DIR/RTDATA0>
```

2. Remove site 4 (mg3cp0) from ATA (11/70 system).

```
RMSITE MG3CP0
```

System response:

```
MG3CP0 FOUND

<RM -F /OFFICE/MG3CP0/*>
<RMDIR /OFFICE/MG3CP0>

<RM -F /RUN.DIR/RTDATA4>
```

Generic 3:

3. Remove site 6 (KEYPORT) from ATA (11/70 system).

```
RMSITE KEYPORT
```

System response:

```
KEYPORT FOUND
RELEASING DEDICATED LOGGING BLOCKS


<RM -F /OFFICE/KEYPORT/*>
RMDIR /OFFICE/KEYPORT>


<RM -F /RUN.DIR/RTDATA6>
```

RTDSAVE
Generic 3:
RTDSAVE sitename

A. Description: The rtdsave command saves the rtdata for a site. When any bin status is altered for a site, the rtdsave program must be executed to save the altered data. If the rtdsave program is not run, altered bin status is lost whenever ATA is started.

B. ***Example:*** **RTDSAVE JTKO**

SACOPY

A. ***Description:*** The stand alone copy command is used to load a copy program into core. In response to the boot prompt message, the operator enters **SACOPY**. The program will prompt the operator for the mode. There are three modes: copy, vrfy, and vcopy.

The copy mode does a copy from one device to another. The vrfy mode verifies the result of the previous copy. The vcopy copies from one device to another, and when the copy is completed, it verifies the copy.

After the mode is entered, the program prompts the user for the source device, <from>. The program then requests the destination device, <to>. Both names are available devices on the system.

If the source or destination device is not found, sacopy prints **INVALID DEVICE** and reissues the <from> or <to> names. The available devices are described below:

rk0—rk device 0
rk1—rk device 1
rp0—first half of an RP03 disk
rp1—second half of an RP03 disk
rputil—first 3400 blocks on an RP03 disk
rf0—rf drive 0—1024 blocks
tm0—tu10 drive 0
tm1—tu10 drive 1
tm4—tu10 drive 0—no rewind
tm5—tu10 drive 1—no rewind
hp0—first quarter of an RP04 disk
hp1—second quarter of an RP04 disk
hp2—third quarter of an RP04 disk
hp3—fourth quarter of an RP04 disk
util—first 4180 blocks of an RP04 disk
hs3—RJS03 disk
hs4—RJS04 disk
ht0—tu16 drive 0—800 bpi
ht1—tu16 drive 1—800 bpi
ht4—tu16 drive 0—800 bpi—no rewind
ht5—tu16 drive 1—800 bpi—no rewind
ht8—tu16 drive 0—1600 bpi
ht9—tu16 drive 1—1600 bpi
ht12—tu16 drive 0—1600 bpi—no rewind
ht13—tu16 drive 1—1600 bpi—no rewind

SACOPY is the best procedure used for performing normal system backup.

B. Examples:

1. To copy from magtape, the system is bootstrapped from the read-only-memory (ROM) bootstrap program. In response to the boot prompt message, the operator enters:

```
SACOPY
```

System response:

```
STAND ALONE COPY/VERIFY PROGRAM

MODE: COPY, VRFY, VCOPY?  <COPY>

NON-VERIFIED COPY
FROM:  <TM0>
TO:  <UTIL>
```

The tape starts to move as the system starts to copy the contents of the tape on TU10/TU16 drive 0 to the RP04/RP06 disk. If no hard errors are encountered, the system prints:

```
    NO OF BLOCKS 4180
    NO BAD SECTORS ON HP4

    NEXT COPY/VERIFY

    MODE:  COPY, VRFY, OR VCOPY?
```

The operator halts sacopy by depressing ENABLE/HALT switch to HALT.

2. If errors are encountered while reading or writing magtape, the operator is asked if processing is to continue. The operator types **Y** if processing is to continue. Any other response is interpreted as no and sacopy restarts.

```
SACOPY
```

System response:

```
STAND ALONE COPY/VERIFY PROGRAM

MODE:  COPY, VRFY, OR VCOPY? <COPY>

NON-VERIFIED COPY
FROM:  <HT0>
TO:  <UTIL>

TAPE <XXX> ERROR
MTCS2:   <NNN>
MTDS:    <NNN>
MTER:    <NNN>
PROCEED?:
```

This message indicates a tape read or write error as described by <XXX>. The <NNN> printed corresponds to the TU10/TU16 magtape registers. Do not continue after a write error.

SH [-tcvi] [NAME (ARG1...ARG9...)]

A. ***Description:*** The shell command is the standard command interpreter. It is the program which reads and arranges the execution of the command lines typed by most users. It may be called as a command to interpret the files of different commands.

Generic 3:

SH [-ecnpstvx] [arg...]

A. ***Description:*** The shell command is the standard command interpreter. It executes commands read from a terminal file.

STARTATA [INIT]
STARTATA [NOINIT]
STARTATA [ERDSAVE]

A. ***Description:*** The startata command performs the following functions:

(1) Checks to see if ATA is already running. If it is, a warning message is printed and startata exits. It will not allow two ATA Systems to run simultaneously.

(2) Verifies that the fixed-head disk is properly mounted as an accessible file system (Generic 2 only).

(3) Initializes several of the dynamic components of the ATA System: locks, semaphores, queues, spooling areas.

(4) If init is used, it initializes the entire ATA data base: ERD, common, bin information for each site, MRS.

(5) If noinit is used, the data files will be used as they were when ATA was last stopped.

(6) If erdsave is used, all records will be cleared out of the MRS, but ERD will remain without trouble records for summary purposes.

(7) Starts the ATA System running.

(8) Monitors the life/death status of the primary ATA processes. If one or more die, it outputs a death message.

B. ***Reference:*** Stopata command.

STOPATA

A. ***Description:*** The stopata command kills the main processes of the ATA System.

B. ***Example:*** The system response from stopata consists of several death messages, a statement as to whether or not common was saved, and a goodbye.

C. ***Reference:*** Startata command.

SYNC

A. Description: The sync command executes the sync system primitive in order to synchronize images of the file system kept by UNIX operating system on disk and in core. If the system is to be stopped, sync ***must*** be called to ensure file system integrity.

SYSANS OFFICENAME KEY

KEY = ASK, OPT, YES, or ALL

A. Description: The sysans command is used to output the answers to the questions asked by the three interactive sysgen programs (askcodes, genopt, and yesno). The key that is entered with the command indicates which answers are being requested. If the key is either ask, opt, or yes, the answers to the questions for the given office will be listed. If the key is all, the answers to the questions will be supplied for the specified office. ♦The user should be logged in as ADMIN to use the sysans command.♦

B. Example: **SYSANS TRAIN5 YES**

System response:
*****YESNO ANSWERS FOR TRAIN5*****

Answers for yesno program are printed out.

TELINIT [ARG]

A. Description: The telinit command is used to direct the actions of init. The telinit program can only be run by the superuser. It signals init via the kill system call to perform the action called for by the 4-character argument.

ARGUMENT	DESCRIPTION
0-6	Places the system in one of the run states; basically, init processes, on each round, only those entries in /etc/lines which have a matching run state. If the system was in single-user state, change the console switches from 7 immediately for protection.
a,b,c	Processes once the /etc/lines entries have a, b, or c respectively, in their run state field. The system run state remains unchanged. This can be used to start a single process independent of a terminal.
Q	Reexamines the /etc/lines files. There is no change in system run state.
S	Switches to single-user state, however, the console switch register must be set to indicate single-user (07).
R	Causes init to overlay itself with /etc/lines, if in single-user state. The console switch register must be set to 0177770 before use of the R argument.

TERMCNTL

A. Description: The termcntl command is used for adding idle line motor control and new line timing to terminals not equipped to handle these functions internally. The program begins by asking for the site name. After receiving a valid site name, the program asks if the terminal is a DATASPEED* 40 terminal set or a TERMINET† terminal set.

> ***Note:*** Idle line motor control is an option on DATASPEED 40 terminals and this program may not be required. If the terminal is not one of the above, then more information is required for the octal representation of the terminal's motor on, motor off sequence. The program asks if the terminal requires new line timing.

B. Examples:

1. Adding idle line motor control to a TERMINET terminal set.

```
TERMCNTL
```

System response:

```
WHICH SITE? <CENTRAL>
USING A DATASPEED 40 THAT REQUIRES IDLE LINE MOTOR
CONTROL AND NEW LINE TIMING (Y-N)? <N>
USING A TERMINET THAT REQUIRES IDLE LINE MOTOR
CONTROL AND NEW LINE TIMING (Y-N)? <Y>
IS YOUR TERMINET OPERATING AT 300 BAUD (Y-N)? <Y>
```

2. Adding idle line motor control to an unpackaged terminal.

```
TERMCNTL
```

System response:

```
WHICH SITE?  <CLINTON>
USING A DATASPEED 40 THAT REQUIRES IDLE LINE MOTOR
CONTROL AND NEW LINE TIMING (Y-N)? <N>
USING A TERMINET THAT REQUIRES IDLE LINE MOTOR CONTROL AND NEW LINE TIMING (Y-N)?
<N>
NOT A PREPACKAGED TERMINAL - I NEED SOME DETAILS.
CORRECT RESPONSE TO REQUESTS FOR MOTOR CONTROL IS A SEQUENCE OF 1 TO 4 OCTAL
NUMBERS SEPARATED BY SPACES
ENTER MOTOR ON SEQUENCE:  <33 101>
ENTER MOTOR OFF SEQUENCE:  <33 105>
IS NEW LINE TIMING REQUIRED (Y-N)? <N>
```

3. Removing idle line motor control from a site.

```
TERMCNTL
```

System response:

```
WHICH SITE? <OFFXB5>
USING A DATASPEED 40 THAT REQUIRES IDLE LINE MOTOR CONTROL AND NEW LINE TIMING
(Y-N)? <N>
USING A TERMINET THAT REQUIRES IDLE LINE MOTOR CONTROL AND NEW LINE TIMING (Y-N)?
<N>
NOT A PREPACKAGED TERMINAL - I NEED SOME DETAILS.
```

*Registered trademark of AT&T.

†Registered trademark of General Electric.

```
CORRECT RESPONSE TO REQUESTS FOR MOTOR CONTROL IS A SEQUENCE OF 1 TO 4 OCTAL
NUMBERS SEPARATED BY SPACES
ENTER MOTOR ON SEQUENCE:  <0>
ENTER MOTOR OFF SEQUENCE:  <0>
IS NEW LINE TIMING REQUIRED (Y-N)? <N>
```

TRANSFER LNXX LNYY (not used in Generic 3)

A. Description: The transfer command is used to transfer line data from one line to another.

B. Example: **TRANSFER LN50 LN60**

System response:

```
SITE WARREN NOW HAS LINE 60
```

C. References: Remove, restore, and line control commands in Section 190-105-301.

/ETC/UMOUNT SPECIAL

A. Description: The umount command announces to the system that the removable file system previously mounted on the special file is to be removed.

If a **MOUNT DEVICE BUSY** message is printed, a process is using the file system, and the file system cannot be unmounted until the file system is free.

B. Example: **/ETC/UMOUNT /DEV/ATADEV**

C. Reference: Mount command.

UNIXSHELL ♦(not used in Generic 3)♦

A. Description: The unixshell command is used to add or remove UNIX shells from a line.

B. Examples:

1. Add UNIX shell to lines 12 and 13.

```
UNIXSHELL
```

System response:

```
DO YOU NEED INSTRUCTIONS?  <NO>


LINE NUMBER ? <12>
LINE 12 IS VACANT.  SHALL I GIVE IT A UNIX SHELL?  <YES>
WHAT SPEED (1200 OR 300) ?  <300>
LINE NUMBER ?  <13>
LINE 13 IS VACANT.  SHALL I GIVE IT A UNIX SHELL?  <YES>
WHAT SPEED (1200 OR 300) ?  <300>
LINE NUMBER ?  <DONE>
LINE FILES UPDATED.
```

```
ACTIVATING CHANGES NOW.
  OK
```

2. Remove UNIX shell from lines 22 and 34.

```
UNIXSHELL
```

System response:

```
DO YOU NEED INSTRUCTIONS?  <NO>

LINE NUMBER ?  <22>
LINE 22 HAS A UNIX SHELL.  SHALL I REMOVE IT?  <YES>
LINE NUMBER ?  <34>
LINE 34 HAS A UNIX SHELL.  SHALL I REMOVE IT?  <YES>
LINE NUMBER ?  <DONE>
LINE FILES UPDATED.
ACTIVATING CHANGES NOW.
  OK
```

3. Add UNIX shell to an ATA line.

```
UNIXSHELL
```

System response:

```
DO YOU NEED INSTRUCTIONS?  <NO>

LINE NUMBER ?  <12>
JFK12 ACTIVE ON LINE 12
LINE NUMBER ?
```

/ETC/UPDFS [-NXCTVIOGF] [NAME] [FILE] [DIR]
▶Generic 3:
/ETC/UPDFS [-NMXCTVIOGFUE] [TAPEFILE] [NAME] [FILE] [DIR]◀

A. Description: The update file system command reads magtapes and updates the file system named by the argument DIR. Its actions are governed by specified flags. These flags are described below:

FLAG	**DESCRIPTION**
N	Designates which magtape drive the program should use.
M	▶Uses the tapefile file as input instead of a magtape drive. (Generic 3 only.)◀
X	Extracts files from the tape and writes them into the file system as given by their name on the tape.
C	Compares the files on the tape with the file system under the argument DIR.
T	Produces a list of files for which UPDFS did something.
V	Produces verbose output.

I — Ignores all files on the tape which have the same relative pathname as one of the pathnames in an ignore file or which are hierarchically lower.

O — Looks at only files which match a name in an only file or which are hierarchically lower.

G — Looks at the tape for the single file name given by the argument NAME.

F — Useful only if the flags X and G have been specified. This flag causes the single file name to be extracted from the tape and given the name FILE.

U — ▶Useful only if the flag X has been specified. This flag unlinks any file about to be created. (Generic 3 only.)

E — Restores the file system to its state when the cmpfs was performed. This option is used with updfs after reloading an epoch. (Generic 3 only.)◀

VRFYLINE [line]

A: Description: The vrfyline command searches the system tables and prints the line data. If the argument line is not used, the status of all lines is printed. The user should be logged in as admin when using this command.

B. Examples:

1. Print the status of line 0.

```
VRFYLINE 0
```

System response:

```
LINE 0    MDT    SITE:    ATA — JFKO    TTY — JFKO
```

▶Generic 3:

2. Print the status of line 0.

```
VRFYLINE 0
```

System response:

```
LINE 0    MPX    SITE:    ATA — JFKO    TTY — JFKO◀
```

VRFYSITE [sitename]

A. Description: The vrfysite command searches the system tables and prints the site data. If the argument sitename is not used, the status of all sites is printed. The user should be logged in as admin when using this command.

B. Examples.

1. Print the site status for all sites

```
VRFYSITE
```

System response:

```
NAME:   JFKO                           TYPE:  XBAR5            SITE:  0
          SPEED:  300                  LINE:  00
          KEYBOARD:  OFF               ATA:  OFF
          ON-SITE DEVICE:  MDT

NAME:   CENO                           TYPE;  XBAR5            SITE:  1
          SPEED:  300                  LINE:  01
          KEYBOARD:  OFF               ATA:  OFF
          ON-SITE DEVICE:  MDT

NAME:   SCC                            TYPE:  XBMOM            SITE:  63
          SPEED:  1200                 LINE:  24
          KEYBOARD:  ON                SITE POWER:  SCC
          MODES:  TASC UNIX ETS        DEFAULT MODE:  UNIX
          TERMINAL TYPE:  DS40-1
```

Generic 3:

2. Print the site status for all sites.

VRFYSITE

System response:

```
NAME:   JFKO                           TYPE:  XBAR5            SITE:  0
          SPEED:  300                  LINE:  00
          KEYBOARD:  ON                ATA:  ON
          ON-SITE  DEVICE:  MDT

NAME:   CENO                           TYPE:  XBAR1            SITE:  1
          SPEED:  300                  LINE:  20
          KEYBOARD:  OFF               ATA:  OFF
          ON-SITE DEVICE:  MDT

NAME:   KEYPORT                        TYPE:  5ETS             SITE:  2
          SPEED:  1200                 LINE:  38
          KEYBOARD:  ON                ATA:  ON
          HISTORY:  ON                 BLOCKS:  1000
          ON-SITE DEVICE:  MDT

NAME:   SCC                            TYPE:  SBMON            SITE:  63
          SPEED:  1200                 LINE:  24
          KEYBOARD:  ON                SITE POWER:  SCC
          TERMINAL TYPE:  DS40-1       MODES:  TASC UNIX ETS
          DEFAULT MODE:  ATA
```

XBTYPE KEY
KEY = 1 or 5
Generic 3:
KEY = 1, 5, e ▶(PDP-11/70 only), or f◀

A. Description: The xbtype command is a shell program that calls the trblcrd, featbuild, ldsos, and ldbin programs.

B. Example: **XBTYPE 5**

System response:

```
BEGINNING TO PROCESS XB5
PROCESSING CARD TYPE 4393
PROCESSING CARD TYPE 3638
PROCESSING CARD TYPE 5488
PROCESSING BINS FOR XB5
FEATURE BUILDING COMPLETE
STAND OFFICE SORTS HEADER COMPLETE
BIN 0 PROCESSED
BIN 100 PROCESSED
BIN 200 PROCESSED
BIN 300 PROCESSED
BIN 400 PROCESSED
BIN 500 PROCESSED
BIN 600 PROCESSED
TOTAL BINS PROCESSED = XXX
BIN 0 PROCESSED
BIN 100 PROCESSED
BIN 200 PROCESSED
BIN 300 PROCESSED
BIN 400 PROCESSED
BIN 500 PROCESSED
BIN 600 PROCESSED
TOTAL BINS PROCESSED = XXX
XBTYPE PROCESSING COMPLETE FOR XB5
```

YESNO SITENAME KEY
KEY = 1 or 5
Generic 3:
KEY = 1, 5, e ▶(PDP-11/70 only), or f◀

A. Description: The yesno command is an interactive program which is used to find out whether the office being generated has certain features. An example is a yes or no answer to the question of whether the office has more than 10 recorders. The answers to the feature questions are used to prune out parts of the analysis I tree which are inappropriate for the office. The answers are left in a features file in the office directory. The /office/features.xb? file is read by the program to determine which feature questions are to be asked.

B. Example: **YESNO TRAIN5 5**

System response:

```
NAMES OF FEATURES WILL BE PRINTED, AFTER EACH FEATURE NAME, TYPE Y OR YES IF FEATURE
EXISTS IN THE OFFICE BEING GENERATED; TYPE N OR NO IF NOT.

NO. 5-CROSSBAR?  <Y>
TCGATCGB?  <N>
MORE-THAN-10-RECORDERS?  <Y>
E5488-CARD?  <N>
OA-AND-OB-PUNCHES-IDENTITY-NUMBER-GROUP-CONN?  <N>
E3638-CARD?  <N>
CENTRAL-PHASE-III?  <N>
RATE-TREATMENT?  <N>
60-CLASSES-OF-SERVICE?  <N>
AIS?  <Y>
NUISANCE-CALLS-TRAPPED-WITH-XFUT?  <N>
NUISANCE-CALLS-TRAPPED-WITH-XTV?  <N>
NUISANCE-CALLS-TRAPPED-WITH-GT5?  <N>
MORE-THAN-10-OFFICE-INDICES?  <N>
CODE-COMPRESSION?  <N>
BILLING-DATA-TRANSMITTER?  <N>
CAMA?  <N>
IDDD-TSPS?  <N>
UNIVERSAL-PAD-CONTROL?  <N>
FEATURE QUESTIONING COMPLETE
```

ZAPTABLES

A. ***Description:*** The zaptables command creates and initializes certain system tables. The command should only be run when the system is being generated from scratch, not when sites are being added.

B. Output Messages

3.02 The output message entry consists of five parts of information which are as follows:

(a) Output message format shows the format for the particular message.

(b) Description gives a detailed explanation of the meaning of the message.

(c) Explanation of variable field identifies any variable fields.

(d) Action to be taken summarizes what should be done in response to the message.

Note: All of the following error messages are detected by the operating system.

BAD BLOCK ON <MAJOR> <MINOR>

A. ***Description:*** The bad block message is printed if, during the deallocation of blocks constituting a file, a block number outside the range of the file system specified by major and minor is encountered.

B. ***Action To Be Taken:*** Begin recovery procedure FDSI.

BAD COUNT ON <MAJOR><MINOR>

A. Description: The bad count message indicates that the core copy of the superblock for the file system specified by major and minor was found to contain information.

B Action To Be Taken: If problem persists, begin recovery procedure FDSI.

GETTY:COULD NOT OPEN line

A. Description: This message indicates getty was unable to open a line which was specified by its first argument when invoked.

B. Action To Be Taken: Begin recovery procedure FDSI.

GETTY:NO ARG

A. Description: This message indicates getty was invoked with no arguments and consequently does not know what line to open.

B. Action To Be Taken: Begin recovery procedure FDSI.

INIT: CAN'T FORK

A. Description: This message indicates an init is unable to generate a new process. This is due to a system overload condition.

B. Action To Be Taken: If this message occurs frequently, begin recovery procedure RA.

INIT:EXEC OF /bin/su FAILED

A. Description: This message indicates init attempts to execute /bin/su when changing to the single user mode to verify the supervisor's password. The message indicates that /bin/su could not be executed.

B. Action To Be Taken: Verify that bin/su program exists and is operating correctly. If not, begin recovery procedure FDSI.

INIT: INIT PROCESS TABLE FILLED (> NUM)

A. Description: This message is generated whenever init is requested to start a new process and its internal process table is full.

B. Action To Be Taken: Reboot.

INIT:enum IS AN INVALID IPROC TABLE ENTRY

A. Description: This message indicates an internal error in init.

B. Action To Be Taken: Begin recovery procedure FDSI.

INIT:NOT SINGLE USER--REBOOT DISALLOWED

A. Description: This message indicates that init was told to perform an overlay of itself when it was not in single user mode.

B. Action To Be Taken: Change to single user mode before asking init to overlay itself.

INIT:OVER num TOKEN LIMIT

A. Description: This message indicates an entry in the lines file has more than num words on it.

B. Action To Be Taken: Begin recovery procedure FDSI.

INIT: OVERLAY WITH /etc/xinit FAILED

A. Description: This message indicates that the init was unable to replace itself with xinit.

B. Action To Be Taken: Begin recovery procedure FDSI.

INIT PROC DEAD

A. Description: This message indicates that the system initialization program (/etc/init), which should never die, has been detected as dead.

B. Action To Be Taken: Although the system may keep running, it should be rebooted. This message should never occur. If it occurs, problems may exist within the operating system.

INIT: PROCESS NOT EXECUTED

A. Description: This message indicates that init has been told to execute the process program but was unable to do so.

B. Action To Be Taken: This is probably due to the program specified in the lines file being nonexistent or nonexecutable. Begin recovery procedure FDSI.

INIT:SYNTAX VIOLATION IN LINES FILE ENTRY AT lid

A. Description: This message indicates the entry in the lines file with lid contains a syntax error.

B. Action To Be Taken: Begin recovery procedure FDSI.

NO FILE

A. Description: This message indicates that a process attempts to open a file when the file table is filled.

B. Action To Be Taken: If problem persists, begin recovery procedure FDSI.

NO SPACE ON <MAJOR><MINOR>

A. Description: This message indicates that there are no more free blocks on the logical device specified by major and minor.

B. Action To Be Taken: Change to run level 6 (Section 190-105-312) and utilize the check command. If the output from this command indicates that there are fewer than 500 free blocks, inform users to delete unnecessary files. If the problem persists and there are more than 500 free blocks indicated by the check command, begin recovery procedure SRZT.

PANIC:NO FS

A. Description: This message occurs when the system cannot find the in-core superblock for a particular device by searching the systems mount table.

B. Action To Be Taken: Begin recovery procedure FDSI.

Generic 3:
PANIC:SWAP ERROR

A. Description: This message indicates that an unrecoverable error occurs when attempting to read or write from the swap device. This message will appear only when the recovery procedure for the respective block device driver fails.

B. Action To Be Taken: Begin recovery procedure FDSI.

Generic 3:
KA6 = <x>
APS = <y>
TRAP TYPE <z>
PANIC:TRAP

A. Description: When a PANIC:TRAP occurs, the messages shown above are printed and the processor goes into a wait state. This panic message indicates that an illegal trap occurred while the processor was executing the operating system. This trap can result from an illegal memory reference by the operating system or the execution of an illegal instruction by the operating system. It can also indicate hardware problems.

B. Variable Field: The number **x** is the value of the kernel data address register at the time of the trap. The number **y** is the address of the saved ps from the trap. The number **z** is an internal number used to determine the cause of the trap. The causes of the trap are shown below:

TYPE	CAUSE
0	Bus error
1	Illegal instruction

2	Breakpoint trace
3	I/O time-out
4	Power fail
5	Emulator trap
6	System entry
7	Programmed interrupt
10	Floating point
11	Segmentation violation.

C. ***Action To Be Taken:*** If the problem persists, suspect hardware problems. Begin recovery procedure FDSI.

PANIC:UNLINK--IGET

A. Description: This message indicates a directory entry exists for the file but inode cannot be brought into core. If a system call to unlink a file is issued, UNIX operating system must bring the inode for that file into core. This removes it from the file system.

B. Action To Be Taken: Begin recovery procedure FDSI.

PARITY
lerr herr mserr mer
PANIC:PARITY (optional)

A. Description: This message occurs when a memory parity error is detected. The four octal numbers **lerr, herr, mserr,** and **mer** are the values contained in the memory system registers located at the UNIBUS* addresses 017777740, 017777742, 017777744, and 017777746 at the time of the error. If the error occurs within the operating system, the optional PANIC occurs.

B. Action To Be Taken: If the error results in a **PANIC:PARITY** message, clear the error from the console before attempting to reboot. Using lerr and herr, determine the address of the word in memory containing the parity error. From the console, deposit a zero into this location. Refer to the 11/70 Processor Handbook for a description of how to interpret the registers. If the problem persists, consult Digital Equipment Corporation (DEC).

♦Generic 3 with PDP-11/40:
PWR FAIL

ATA-UNIX RELEASE 1.1
MEM = XXX
DEV ROOTDEV
PHASE 1 - CHECK BLOCKS
PHASE 3 - CHECK PATHNAMES
PHASE 4 - CHECK REFERENCE COUNTS ♦

*Registered trademark of Digital Equipment Corporation

```
▶ UNREFERENCED FILE  I = XXXX CLEAR?  NO
  UNREFERENCED FILE  I = XXXX CLEAR?  NO
  UNREFERENCED FILE  I = XXXX CLEAR?  NO
  PHASE 5 - CHECK FREE LIST
  XXXX FILES   XXXXX BLOCKS   XXXX FREE

  LOGIN:
```

XXX = Logger, Analysis I, Immediate Action, Analysis II, Report Gen, Queue Scanner, Audits, Kronos, MDThdlr, Atash, RRpurge, TASC-ATA Daemon, Read Load Pipe, or Error Logger

A. Description: When a power fail message occurs, UNIX operating system is restarting. A check is done and printed on the console to give an idea of the file system sanity. If a check fails, it is automatically answered no. If ATA was running when the power failed, it is restarted and the respective program identifications (PIDs) of the ATA process are printed out.

Generic 3 with PDP-11/70:

```
668

*** POWER FAIL RESTART ***

--- POWER FAIL ATA RESTART ---

LOGIN:
LOGIN:
DEV   ROOTDEV
PHASE 1 - CHECK BLOCKS
PHASE 3 - CHECK PATHNAMES
PHASE 4 - CHECK REFERENCE COUNTS
PHASE 5 - CHECK FREE LIST
XXXX  FILES   XXXXX  BLOCKS    XXXX  FREE
```

A. Description: The number 668 indicates the number of machine cycles that occurred from the time power fail was detected until the machine stopped running. The first power fail message is issued by UNIX operating system and the second power fail message is issued by ATA. The LOGINs occur when the power failure monitor program assures that all processing is stopped. A check is done and printed on the console to give an idea of the file system sanity. If a check fails, it is automatically answered no. If ATA was running when the power failed, it is restarted and the respective PIDs of the ATA process are printed out.

If the system was at run level 6 when the power fail occurred, the two power fail messages are the only information printed. The system is then restored to run level 6.◀

RANDOM INTERRUPT IGNORED

A. Description: This message indicates a trap is taken through location zero. This indicates a hardware problem.

B. Action To Be Taken: If the problem persists, consult DEC.

RP04/5/6 DRIVE <n> OFFLINE

A. Description: This message indicates that an RP04, RP05, or RP06 drive is switched offline. If this message occurs when no manual action has been taken at the drive, a hardware problem is indicated.

B. Action To Be Taken: When this message is generated, all pending I/O for the offline drive is flushed.

RP04/5/6 DRIVE <n> OFFLINE. MAN ATTN.

A. Description: This message indicates an RP04, RP05, or RP06 drive does not come back online after a power failure. The message appears every 16 minutes until the offending drive is placed back online.

B. Action To Be Taken: Ensure that the drives specified are powered on and online.

STRAY INTERRUPT AT addr

A. Description: This message indicates an unexpected interrupt occurs through the octal address addr.

B. Action To Be Taken: If the problem persists, the offending hardware should be isolated and repaired.

UNIT num UNLOADED

A. Description: This message indicates a severe or persistent error condition.

B. Action To Be Taken: Repair the offending drive. Refer to the error log to determine the type of error causing the drive to be placed offline.

---

4. RECOVERY PROCEDURES

4.01 The recovery procedures are designed to recover the system from software detected errors which may exist in the ATA hardware or software. The recovery procedures are used to attempt to restore the system to a software configuration which is known to have functioned correctly. If this configuration is restored but the system continues to fail, hardware problems may be indicated. Each succeeding procedure attempts to recover the system by taking a more drastic action than the procedure immediately before it. This results in a greater impact on service.

Note: In all cases, consider the impact of the trouble condition.

A. Partial Dynamic System Initialization (PDSI)

1. ***Procedure:***

Note: If the program generating the message is an interactive (user entered) command, repeat the command before proceeding. If the trouble is isolated (a single message or messages generated by a single program) and is not affecting basic system functions, this procedure may be postponed until a convenient time.

a. Ask all users to log off the system.

b. Change to run level 6 (Section 190-105-312).

c. Change to run level 0 (Section 190-105-312).

d. Determine if problem persists; if so, go to next procedure.

2. ***Duration:*** 2 to 3 minutes.

3. ***Impact on System Functions:*** Almost all system functions will be terminated and restarted. There should be little or no loss of user or office data residing on disk.

B. Full Dynamic System Initialization (FDSI)

1. ***Procedure:***

Note: If the problem generating the message is an interactive (user entered) command, repeat the command before proceeding. If the trouble is isolated (a single message or messages generated by a single program) and is not affecting basic system functions, this procedure may be postponed until a convenient time.

a. Stop the system gracefully (Section 190-105-312).

b. Boot the system from disk (Section 190-105-312). When run level 6 is attained, check the file system integrity.

c. Change to run level 0 (Section 190-105-312).

d. Determine if problem persists; if so, go to next procedure.

2. ***Duration:*** 5 to 10 minutes.

3. ***Impact on System Functions:*** The core memory of the system will be reinitialized and all system functions will be terminated and restarted. There should be little or no loss of user or office data residing on disk. All incoming ATA messages will be lost during the initialization period.

C. System Restoral From Backup Tapes (SRBT)

1. ***Procedure:***

a. Obtain the latest epoch and incremental backup tapes. If the date on the latest incremental backup tape indicates that it was taken prior to the latest epoch backup, use only the epoch backup tape.

b. Restore the primary file system (Section 190-105-312).

c. Determine if the problem persists; if so, go to the next procedure.

d. Warn all users that they may have lost some data files and ATA data.

2. ***Duration:*** 30 minutes.

3. ***Impact on System Functions:*** Disk memory will be reinitialized to the system state at the time the epoch and incremental backups were taken. Core memory will be reinitialized and all system functions will be restarted. All ATA messages received during the restoral will be lost. Messages previously logged onto disk will normally not be lost. Depending on the age of the backup tapes, some user and office data may be lost. This data should be reentered as required if system recovery is successful.

D. System Restoral From Day Zero Tape (SRZT)

1. ***Procedure:***

   a. Restore the primary file system using the day zero epoch backup tape (Section 190-105-312).

   b. Determine if the problem persists; if so, go to the next procedure.

   c. Warn all users that they may have lost all of their data files and ATA data.

2. ***Duration:*** 30 minutes.

3. ***Impact on System Functions:*** Disk memory will be reinitialized to the system state at the time the day zero tape was created. Core memory will be reinitialized, and all system functions will be restarted. All ATA messages received during the restoral will be lost.

E. Request Assistance (RA)

1. ***Procedure:***

   a. Contact the Western Electric regional engineer and explain the problem.

   b. Attempt to continue operations with the faulty system unless it appears that the cause is a major hardware problem.

   c. If the resolution of the problem appears to be a program design error, submit a trouble report. The trouble report should include as much data as possible about the circumstances surrounding the error.

2. ***Duration:*** Unknown.

3. ***Impact on System Functions:*** Unknown.